



Development Tools

Getting Started

with Freescale software and tools
for automotive microcontrollers



Did you know?

Freescale offers starter kits for many devices that include the software development tools, cables, a wiggler (debug interface), an MCU and a board needed to begin development.

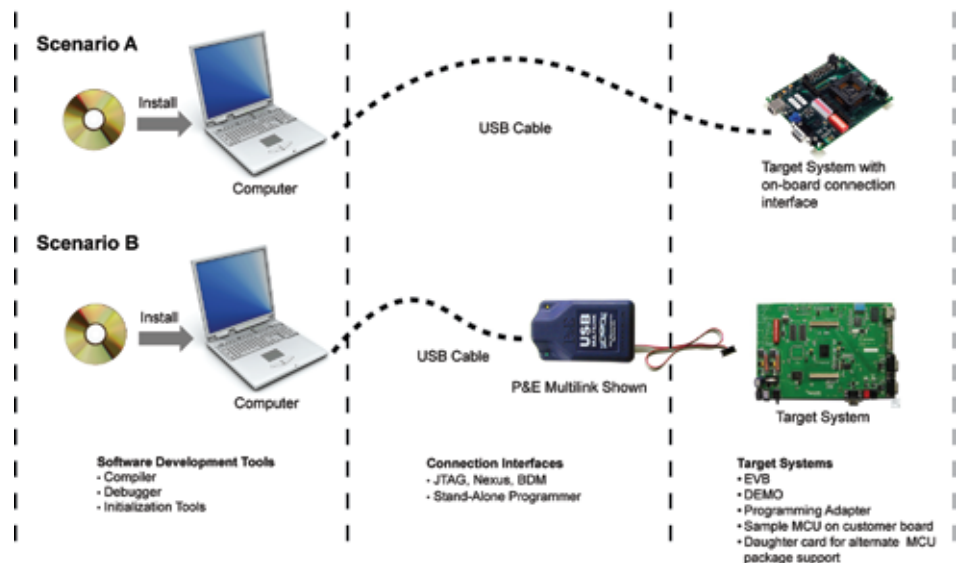
A Primer on the Development Process

Once a microcontroller (MCU) is deployed for production, it may be inconvenient, cost prohibitive or even impossible to modify. Therefore, you should ensure that the final MCU program (or run-time software) meets application requirements. You need development tools with sufficient features, capabilities, configurability and flexibility to facilitate the development of final production run-time software for the embedded MCU.

This brochure offers an overview of the development process and covers different aspects of run-time software. With this understanding, you can quickly create your own development environment.

The Three Primary Components of a Development Environment

- **Software development tools** are applications residing on a host computer that are used for writing, compiling and debugging code. Additional applications may include initialization and code generation tools. Depending on device architecture, Freescale offers Processor Expert™ and RAppID tools to support initialization.
- **Connection interfaces** bridge the gap between the code developed on a computer and the target system. For Freescale, the connection interface provides two key functions in the development environment, specifically, the ability to flash program the target system and provide a link to the target's on-chip debug hardware.
- **Target system** is the MCU. In the absence of an MCU, the target system can be an emulator or simulator.



Examples of the software development process

Write, Compile, Debug

With the development environment in hand and the software development tools installed on the host computer, the next task is to develop a bug-free application for the end product. To achieve this, development becomes an iterative process of creating code and verifying its operation. There are several steps in the development process:

- Write/edit source code
- Compile and link code to develop an executable for the target system
- Flash executable onto a target system
- Validate the application and debug the code
- Repeat steps above if necessary

More about Debugging

Debugging utilizes one or more dedicated pins on the MCU to access built-in debugging hardware. The on-chip debugging hardware typically provides the ability to read and write memory, I/O and the CPU registers, and to start and stop program execution. In addition, the target MCU generally contains a dedicated logic block that implements a few hardware breakpoints. While not as flexible and powerful as an in-circuit emulator, it can provide very transparent access to the MCU and target system. In addition, the connection between a host computer and the target MCU is facilitated by an inexpensive piece of hardware known as a wiggler.

In-Circuit Emulation

An in-circuit emulator uses the actual target device in an expanded mode so that external circuitry can monitor both internal and external bus activity. Because some of the microcontroller's I/O ports are lost when placing the device in expanded mode, an emulator will typically utilize an FPGA to replace the lost ports. Additionally, the external circuitry typically provides a virtually unlimited number of complex hardware breakpoints and large amounts of bus trace memory. In addition to source-level debugging, many emulators provide additional software analysis tools, such as code coverage and performance analysis.

Pre-Silicon Emulation

Before an announced MCU device is physically available, pre-silicon emulation can be used to mimic the behavior of that MCU with other hardware, usually a special purpose pre-silicon emulator system. The tool allows an entire system to be debugged before the silicon is available. The emulator is typically plugged into a target system in place of the MCU, which allows a developer to test software in real time in a target system environment.



Did you know?

Did you know that Freescale offers factory programming? Learn more at www.freescale.com/factoryprogramming.

Simulation

Even without silicon you can start development using Freescale's simulation targets. They allow developers to analyze how MCUs and their programs will act in an application environment. The simulator shows the internal processor state and that of the outputs and allows input signals to be generated. Simulators can be the quickest way to debug and analyze problems.

Understanding Run-Time Software Resources

The embedded MCU development process is designed to create and validate binary executables that run on the embedded MCU once deployed. This application run-time software can be developed from scratch using custom source code, from off-the-shelf prewritten source code or through a combination of both.

Run-time software resources can provide you with a starting point for application development or, in some cases, with the total solution. Off-the-shelf prewritten software generally includes:

- **Operating systems**, which manage CPU resources, I/O, communication and synchronization constructs, processes, threads and tasks.
- **Device drivers, libraries and protocol stacks**
 - o Device drivers are target run-time software used to initialize and drive specific SOC or hardware board peripherals, such as Ethernet, USB and serial.
 - o Libraries are run-time software components shipped as a package and intended for re-use.
 - o Protocol stacks are communication protocols and other software that typically sit logically on the target OS. They include software used to communicate between disparate processes (e.g. TIPC or CORBA) or software used to communicate with other devices, such as Ethernet or USB.

Production Programming Options

In cases where numerous MCUs need to be flashed with final code for mass production and duplication, developers have three options:

- **Stand-alone programmer** allows you to connect directly to a board to program, test or debug an MCU.
- **Gang programmer** is a device with slots for programming several MCUs at once.
- **Factory programming** is where a customer provides final code to an MCU supplier for code installation onto the MCUs during production.

Getting Our Tools Into Your Hands

Most online tools listings will include a [Buy Direct](#) button and/or a [Distributor](#) button. Click “Buy Direct,” and you will start the process of buying your selected tool directly from Freescale. Click “Distributor,” and a pop-up will include links to the Freescale authorized distributors that stock the tool you’ve chosen.

CodeWarrior™ Suites

The CodeWarrior development tools are sold in suites. Purchasing a suite entitlement enables you to register for any or all of the CodeWarrior products contained within that suite. This allows you to gain access to Freescale’s entire portfolio of CodeWarrior tools across the suite you purchase.

CodeWarrior Suites can be purchased in a variety of configurations, including node-locked or floating licenses and 12-month renewable or permanent licenses.

Start your development and evaluation with CodeWarrior special edition, which is available at no charge.

The CodeWarrior Development Studio for Microcontrollers, (Special Edition) is available at no charge. It allows you to develop projects with unlimited assembly code, up to 32 KB of C code for HC(S)08.

The CodeWarrior Development Studio for HCS12(X) Microcontrollers, (Special Edition) is available at no charge and allows you to develop projects with unlimited assembly code, up to 32 KB of C code and 1 KB of C++ code for HCS12(X) derivatives, and up to 512 bytes of C or C++ code for the XGATE controller.

Processor Expert™

CodeWarrior development suites can also come with Processor Expert, a rapid application development (RAD) component-based application builder and system integration tool. This tool enables component-oriented programming, provides tested, ready-to-use code and delivers instant functionality of generated code.

Find out more about the CodeWarrior development suites and additional supporting products, including value-added extensions and run control devices, by visiting freescale.com/codewarrior.





RAppID

RAppID graphical development tools for the 32-bit Power Architecture® microcontrollers enable users to quickly and easily configure the controller plus generate complete documentation on this configuration. They can also be used as a learning tool to gain an understanding of the controller and its peripherals. RAppID not only generates C code for initializing the registers, but also provides a system initialization function that brings the controller up in an orderly sequence. Use RAppID to save time and become an expert on the 32-bit Power Architecture families.

OSEK/VDX

Freescale OSEKturbo OS is the leading OSEK/VDX™ Real Time Operating System (RTOS) implementation available on the market today in terms of quality and performance. OSEKturbo fully conforms to the OSEK/VDX Operating System standard. The RTOS is fully configurable and statically defined within the CodeWarrior IDE environment or standalone. Only those functions that are required are included, providing the user with a truly scalable RTOS. OSEKturbo OS is available for a wide range of 8-, 16- and 32-bit embedded microprocessors. The emphasis in the design of OSEKturbo OS was to meet demanding performance requirements. The result is that Freescale OSEKturbo is one of most widely used OSEK/VDX OS implementations on the market today.

AUTOSAR

Freescale's production intent AUTOSAR-compliant software for electronic control units that uses our 16-bit S12X microcontrollers (MCUs) and 32-bit MCUs. Built on Power Architecture technology AUTOSAR is a solution to the soaring software content in automotive system designs and ever-tightening development deadlines.

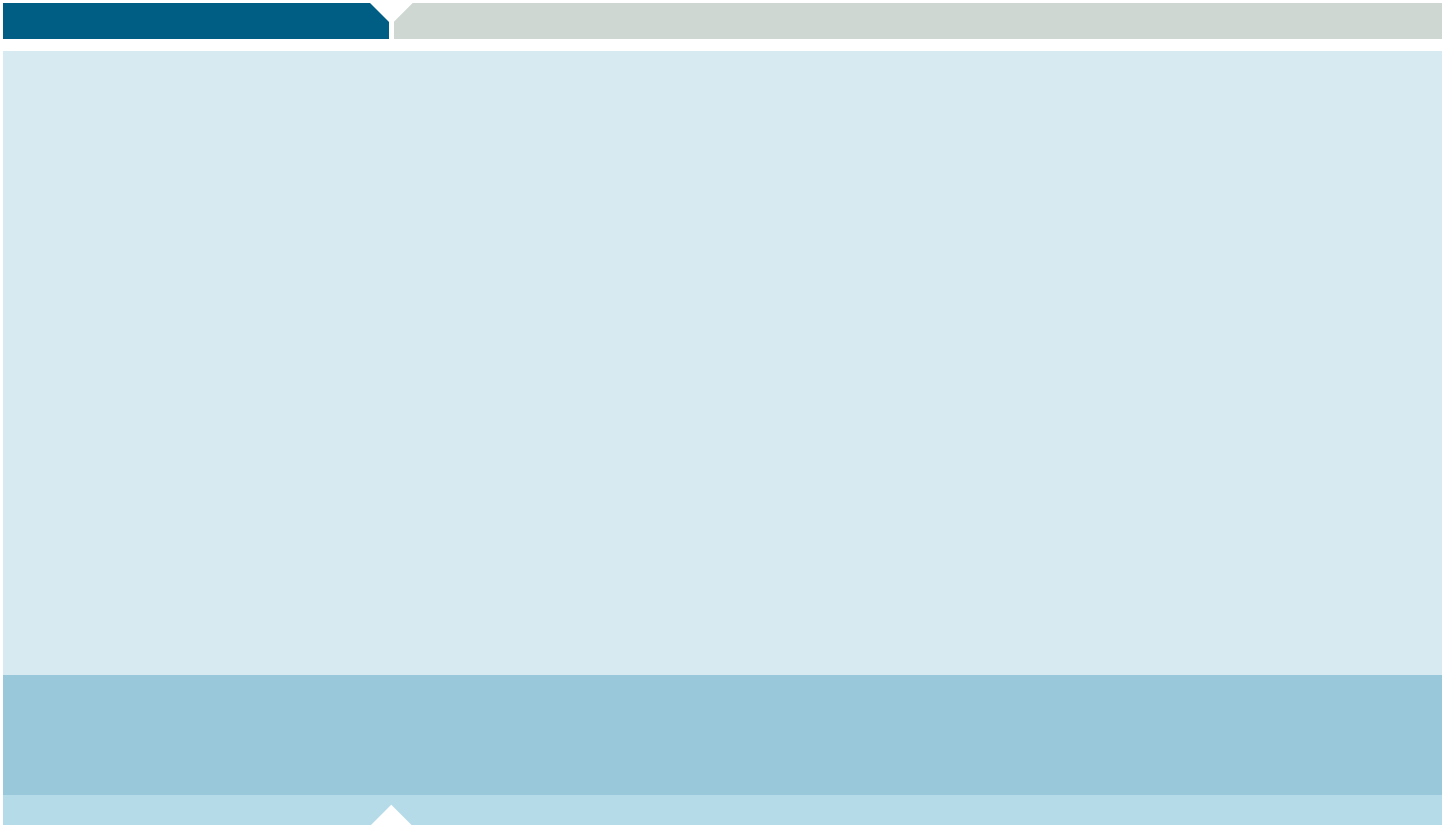
AUTOSAR software addresses the challenge of rising code complexity by providing open automotive software architecture, jointly developed by automobile OEMs, suppliers and tool developers. AUTOSAR supports the development of innovative yet standardized electronic systems that improve quality, performance, safety and environmental friendliness. It also helps simplify the process of updating software and hardware over the life of a vehicle.

Tools Alliance Program

Freescale's Tools Alliance Program (TAP) is a worldwide network of development tools vendors who provide a variety of solutions for our customers. By forming this alliance with industry-leading development tools suppliers, Freescale helps bring the right solution to you. A variety of options are available at www.freescale.com/codewarrior or from individual product summary pages.

The table below contains information on our third-party vendors.

Freescale Vendors for Automotive Microcontrollers																							
Vendor	Web Site	Development Tools						Connection Interface and Targets						Run-Time Software				Technologies and Specialized Tools					
		Build Tools		SW Debugger Tool		Initialization Tool/Code Generation		Connection Interface Tools		Emulators and Evaluation Boards		Simulation		Operating Systems		Drivers, Protocol Stacks, Translators							
ASH WARE Inc.	www.ashware.com			32																			eTPU compiler/simulator
Axiom Manufacturing	www.axman.com									8	16	32	8	16	32								
Byte Craft Limited	www.bytecraft.com	8		32																			eTPU compiler, MCU compiler
CodeSourcery	www.codesourcery.com			32			32																GNU Toolchains
Cosmic Software	www.cosmic-software.com	8	16	32	8	16	32																
CoWare																		32					
eCosCentric	www.ecoscentric.com																			32			
Elektrobit	www.elektrobit.com																				8	16	32
Freescale Semiconductor	www.freescale.com	8	16	32	8	16	32	8	16	32	8	16	32	8	16	32	8	16	32	8	16	32	AUTOSAR MCAL, OSEK, AutoSar OS, eTPU2 libraries, LIN/CAN/FlexRay® drivers, RappID
GreenHills Software	www.ghs.com			32								32						32		32			
IAR Systems	www.iar.com	8	16		8	16																	
iSystem	www.isystem.com				8	16	32			8	16	32	8	16	32								
Lauterbach	www.lauterbach.com				8	16	32			8	16	32	8	16	32								
P&E Microcomputer Systems	www.pemicro.com				8	16	32			8	16	32	8	16	32								
PHYTEC	www.phytec.com														32								
Segger	www.segger.com																					32	Graphics
Simuquest	www.simuquest.com								16														
Treck	www.treck.com																					32	TCP/IP
VaST	www.vastsystems.com																	32					
Vector Informatik	www.vector.com																				8	16	32
Wind River	www.windriver.com			32			32					32						32		32			



Learn More:

For current information about Freescale automotive products and documentation, please visit www.freescale.com/automotive.

